



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/666,343	09/19/2003	Sailesh Kottapalli	42P17404	8176

8791 7590 07/17/2007
BLAKELY SOKOLOFF TAYLOR & ZAFMAN
1279 OAKMEAD PARKWAY
SUNNYVALE, CA 94085-4040

EXAMINER

LI, AIMEE J

ART UNIT	PAPER NUMBER
----------	--------------

2183

MAIL DATE	DELIVERY MODE
-----------	---------------

07/17/2007

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/666,343	Applicant(s) KOTTAPALLI, SAILESH	
	Examiner Aimee J. Li	Art Unit 2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 25 April 2007.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1,2,6-11,15-19,22-25,28 and 29 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-2, 6-11, 15-19, 22-25, and 28-29 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. Claims 1-2, 6-11, 15-19, 22-25, and 28-29 have been considered. Claims 1, 10, 18, and 24 have been amended as per Applicants' request. Claims 4 and 13 have been cancelled as per Applicants' request.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as filed 25 April 2007.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 1-2, 10-11, 17-19, and 24-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Morrison, U.S. Patent Number 6,170,052 (herein referred to as Morrison) in view of Rodgers et al., U.S. Patent Number 6,496,925 (herein referred to as Rodgers).

5. Referring to claims 1, 10, 18, and 24, taking claim 1 as exemplary, Morrison has taught a method, comprising:

- a. Decoding an original instruction into a complementary-predicated pair of instructions (Morrison Abstract "...generating pairs of conditional instructions corresponding to special predicate sequences from single instructions having a predicate..."; column 3, lines 6-25 "...first conditional micro-op belongs to a special predicated sequence. The second conditional micro-op belongs to the

same special predicated sequence...”; column 4, lines 11-15 “...special predicated sequences includes paired conditional instructions that update a destination register regardless of the truth or falsity of the predicate...”; column 4, lines 22-30 “...The special predicated sequence **30** also includes first and second conditional micro-ops **32**, **33**. The conditional micro-ops **32**, **33** are executed when the condition P_x is true and false, respectively...”; column 5, lines 20-28 “...decoders **54**, **56** translate one or more types of conditional instruction into the special predicated sequence, including the predicate evaluating micro-op **31** and the associated pair of conditional micro-ops **32**, **33**...”; Figure 3; Figure 4; and Figure 5, elements 54 and 56) by sending a hint to a register renaming circuit (Morrison column 6, lines 5-23 “...uses one bit of the binary machine word 82 of a micro-op to determine whether the micro-op is the second conditional micro-op 33 of a pair associated with a special predicated sequence 30...”; column 6, lines 46-60 “...using the condition bit **86** of the machine word **82** for a micro-op to identify the second conditional micro-op **33** of a special predicated sequence **30**...”; Figure 7, elements 86, 90, and 94; and Figure 8, element 114), the complementary-predicated pair of instructions including a predicate-positive instruction and a predicate-negative move instruction (Morrison Abstract “...generating pairs of conditional instructions corresponding to special predicate sequences from single instructions having a predicate...”; column 3, lines 6-25 “...first conditional micro-op belongs to a special predicated sequence. The second conditional micro-op belongs to the same special predicated sequence...”;

column 4, lines 11-15 "...special predicated sequences includes paired conditional instructions that update a destination register regardless of the truth or falsity of the predicate..."; column 4, lines 22-30 "...The special predicated sequence **30** also includes first and second conditional micro-ops **32, 33**. The conditional micro-ops **32, 33** are executed when the condition P_x is true and false, respectively..."; column 4, lines 41-57 "...specific embodiments may employ different types of conditional micro-ops and different numbers of conditional micro-ops...conditional micro-op **33** that restores a data word of the register Z in the same register Z is the third micro-op in the special sequence..."; Figure 3; and Figure 4 – In regards to Morrison, the exact instruction for the false predicate micro-op does not matter as long as the original data is restored, e.g. moved, into the correct physical register. In Morrison's example in Figure 3 and 4, an ADD instruction is used to move the data, since the original data is added by 0, making it equal to itself, and the result is stored into the destination register Z.);

- b. Sequencing the predicate-positive instruction and the predicate-negative move instruction for out-of-order execution (Morrison column 2, lines 42-67 "...renaming removes artificial dependencies between instructions such that out-of-order execution does not change the results..."; column 5, lines 39-45 "...one or more execution units **62, 64**...Execution preserves dependencies but not necessarily the original instruction order from the fetcher **52** or decoders **54, 56**..." and Figure 5, elements 62 and 64);

- c. Renaming, by the register renaming circuit, both a first destination register of the predicate-positive instruction and a second destination register of the predicate-negative move instruction to a same physical register, wherein the hint permits renaming the first and second destination registers to the same physical register (Morrison column 3, lines 6-25 "...renamer to rename the destination registers of two conditional micro-ops with a single physical register in response to the two conditional micro-ops having the same destination register and belonging to a single special predicated sequence..."; column 4, line 58 to column 5, line 16 "...a physical destination register Z_1 replaces the logical destination register Z of the original special predicated sequence. In the renamed special predicated sequence **36**, the destination address of both renamed conditional micro-ops **38**, **39** is the same physical register Z_1 ..."; column 5, lines 29-38 "...renamer **58** recognizes paired conditional micro-ops associated with special predicated sequences and performs register renaming in a manner that is consistent..."; column 6, lines 5-22 "...decoders **54**, **56** initialize the condition bit **86** to have a value of logic 1 or logic 0 so that the renamer **58** can identify second conditional micro-ops **33** of special predicated sequences **30**."; column 6, line 46 to column 7, line 10 "...At block **114**, whether the micro-op is a second conditional instruction **33** of a special predicated sequence **30** is determined from the value **90**, **94**, i.e. logic 1 or logic 0, of the condition bit **86**...At block **118**, a physical register assigned to the logical destination register of the micro-op is looked up in the rename table **60**, and assigned the same physical register is assigned for the

logical destination register of the micro-op in response to determining, at block 114, that the micro-op is a second conditional micro-op...”; Figure 4; Figure 5, element 58; Figure 7, elements 86, 90, and 94; and Figure 8, elements 114 and 118) and

- d. Retiring either the predicate-positive instruction or the predicate-negative move instruction responsive to a predicate value associated with both instructions (Morrison column 5, lines 8-16 “A dependent micro-op 34 with a source address that is a destination address of the conditional pair...a dependent micro-op 41 points to the same physical renamed destination register.”; column 5, lines 43-45 “Execution preserves dependencies but not necessarily the original instruction order...”; Figure 3; and Figure 4 – In regards to Morrison, one of the predicate pair instructions must be retired for the dependent instruction to be executed, since dependency is maintained in the system. Dependent instructions will only execute when the instructions they are dependent on retire, i.e. when the instruction is complete and the results are stable and available.).

6. Morrison has not explicitly taught sending the hint via a trace cache. However, Morrison has taught that the dependencies between the instructions are preserved (Morrison column 5, lines 43-45 “Execution preserves dependencies but not necessarily the original instruction order...”), but has not taught the details of preserving the dependencies nor how it effects instruction scheduling. Rodgers has taught that a trace cache contains pointers for preceding and proceeding instructions, e.g. showing dependencies among instructions, and how it effects instruction scheduling (Rodgers column 7, lines 20-62 “...each entry including pointers to

Art Unit: 2183

preceding and proceeding microinstructions comprising the trace...” and Figure 2, element 62). Rodgers shows his trace cache and trace delivery engine is found after the instruction translation engine, i.e. decoder, and prior to the register renamer, and controls which decoded instructions are issued to the back-end of the unit, which contains the renamer. A person of ordinary skill in the art would have recognized, and as taught by Rodgers, the trace delivery engine, which includes the trace cache, increases instruction bandwidth (Rodgers column 7, lines 31-32 “...to provide increased micro-instruction bandwidth...”) and facilitates high-performance instruction sequencing (Rodgers column 7, lines 38-42 “...the trace cache 62 facilitates high-performance sequencing...”). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the trace cache of Rodgers in the device of Morrison to increase instruction bandwidth and facilitate high-performance instruction sequencing.

7. Claims 10, 18, and 24 have similar limitations to claim 1 and are rejected for similar reasons. Claims 10 and 18 differ only in that they are processor claims as opposed to the method of claim 1. Claim 24 differs only in that it is a system claim, which is taught by Morrison (Morrison Abstract “Systems, apparatus, and methods...” – In regards to Morrison, the I/O devices and communication device coupled to a bus are inherent to a computer system. Please see the definition of “computer” provided.).

8. Referring to claims 2, 11, 19, and 25, taking claim 2 as exemplary, Morrison has taught the method of claim 1, wherein the predicate-negative move instruction is responsive to a complement of the predicate value (Morrison Abstract “...generating pairs of conditional instructions corresponding to special predicate sequences from single instructions having a

Art Unit: 2183

predicate...”; column 3, lines 6-25 “...first conditional micro-op belongs to a special predicated sequence. The second conditional micro-op belongs to the same special predicated sequence...”; column 4, lines 11-15 “...special predicated sequences includes paired conditional instructions that update a destination register regardless of the truth or falsity of the predicate...”; column 4, lines 22-30 “...The special predicated sequence **30** also includes first and second conditional micro-ops **32, 33**. The conditional micro-ops **32, 33** are executed when the condition P_x is true and false, respectively...”; column 5, lines 20-28 “...decoders **54, 56** translate one or more types of conditional instruction into the special predicated sequence, including the predicate evaluating micro-op **31** and the associated pair of conditional micro-ops **32, 33**...”; Figure 3; Figure 4; and Figure 5, elements 54 and 56 – In regards to Morrison, the value opposite to the condition laid forth in the original instruction, i.e. when the condition is false, is the complement value.).

9. Claims 11, 19, and 25 have similar limitations to claim 2 and are rejected for similar reasons. Claims 11 and 19 differ only in that they are processor claims as opposed to the method of claim 2. Claim 25 differs only in that it is a system claim.

10. Referring to claim 17, Morrison has taught the processor of claim 10, further comprising execution units to execute the predicate-positive instruction and the predicate-negative move instruction in parallel (Morrison column 1, lines 14-49 “...No-wait sequential fetching enables the processor reduce the number of time periods when execution units remain idle due to no instruction being available for execution...” and Figure 5, elements 62 and 64 – In regards to Morrison, there are multiple execution units shown in Figure 5 in Morrison, so the predicate-positive instruction and predicate-negative instruction are executed in parallel, when there are no

Art Unit: 2183

other instructions available to execute in parallel to reduce the number of idle time periods experienced by the execution units.).

11. Claims 6-9, 15-16, 22-23, and 28-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Morrison, U.S. Patent Number 6,170,052 (herein referred to as Morrison) as applied to claims 1, 10, 18, and 24 above, and further in view of Sprangle and Patt's "Facilitating Superscalar Processing via a Combined Static/Dynamic Register Renaming Scheme" ©1994 (herein referred to as Sprangle).

12. Referring to claims 6, 15, 22, and 28, taking claim 6 as exemplary, Morrison has not explicitly taught the method of claim 1, further comprising squashing the predicate-negative move instruction when the predicate-positive instruction executes before the predicate-negative move instruction and the predicate value is true. However, Morrison has taught register renaming that renames associated predicate instructions to the same physical registers during predicate execution but not the details of the actual instruction execution and scheduling.

Sprangle has also taught register renaming that renames associated predicate instructions with the same physical identifiers during predicate execution with the details of the actual instruction execution and scheduling. Sprangle has explicitly taught squashing the predicate-negative move instruction when the predicate-positive instruction executes before the predicate-negative move instruction and the predicate value is true (Sprangle Section 6 "...Additional enabling logic is added to the node tables that allows a predicated instruction to be scheduled to a functional unit after the predicate is evaluated and it is determined that the instruction should execute. For example, if the instructions in table 6 are issued out of basic block (0001), and the predicate P=TRUE, the second instruction satisfies the tag associated with (I)R@=(0001)00010, and the

Art Unit: 2183

third instruction is not allowed to be scheduled to the functional units...”; Table 4; Table 5; and Table 6). A person of ordinary skill in the art at the time the invention was made, and as taught by Sprangle, would have recognized that the instruction scheduling and execution of Sprangle eliminates the need for noops associated with the false predicate, and the extra sources associated with the noop (Sprangle Section 6, paragraph 6 “...By mapping both results to the same tag, and allowing only one instruction to execute, the noop action needed for a conventional ISA is eliminated, along with the extra source associated with the noop action.”), thereby improving the memory and processing efficiency of the system, since the extra space in memory to store the noop instructions and processor resources to execute the noop instructions are no longer wasted.

13. Claims 15, 22, and 28 have similar limitations to claim 6 and are rejected for similar reasons. Claims 15 and 22 differ only in that they are processor claims as opposed to the method of claim 6. Claim 28 differs only in that it is a system claim.

14. Referring to claim 7, Morrison in view of Sprangle has taught the method of claim 6, wherein the squashing occurs before the predicate-negative move instruction executes (Sprangle Section 6 “...Additional enabling logic is added to the node tables that allows a predicated instruction to be scheduled to a functional unit after the predicate is evaluated and it is determined that the instruction should execute. For example, if the instructions in table 6 are issued out of basic block (0001), and the predicate P=TRUE, the second instruction satisfies the tag associated with (I)R@=(0001)00010, and the third instruction is not allowed to be scheduled to the functional units...”; Table 4; Table 5; and Table 6).

15. Referring to claims 8, 16, 23, and 29, taking claim 8 as exemplary, Morrison in view of Sprangle has taught the method of claim 1, further comprising squashing the predicate-positive

Art Unit: 2183

instruction when the predicate-negative move instruction executes before the predicate-positive, instruction and the predicate value is false (Sprangle Section 6 "...Additional enabling logic is added to the node tables that allows a predicated instruction to be scheduled to a functional unit after the predicate is evaluated and it is determined that the instruction should execute... When the predicate P=FALSE, the third instruction satisfies the same tag, (0001)00010, and the second instruction is not allowed to execute..."; Table 4; Table 5; and Table 6).

16. Claims 16, 23, and 29 have similar limitations to claim 6 and are rejected for similar reasons. Claims 16 and 23 differ only in that they are processor claims as opposed to the method of claim 8. Claim 29 differs only in that it is a system claim.

17. Referring to claim 9, Morrison in view of Sprangle has taught the method of claim 8, wherein the squashing occurs before the predicate-positive instruction executes (Sprangle Section 6 "...Additional enabling logic is added to the node tables that allows a predicated instruction to be scheduled to a functional unit after the predicate is evaluated and it is determined that the instruction should execute... When the predicate P=FALSE, the third instruction satisfies the same tag, (0001)00010, and the second instruction is not allowed to execute..."; Table 4; Table 5; and Table 6).

Response to Arguments

18. Applicant's arguments filed 25 April 2007 have been fully considered but they are not persuasive. Applicant's argue in essence on pages 8-9

Applicant submits that Rodgers may not be considered as prior art precluding patentability of the present application...Therefore, due to the §102(e) status, and

in light of §103(c), Rodgers cannot be used as a reference to preclude the patentability...

19. This has not been found persuasive. Rodgers et al., U.S. Patent Number 6,496,925, was published 17 December 2002. The instant application has an effective filing date of 19 September 2003. As such, Rodgers date qualifies as a reference under §102(e) *and* §102(a), because it “was...patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.” Exclusion of a reference under §103(c) requires that the prior qualifies “...*only* under one or more subsections (e), (f), and (g) of section 102 of this title (emphasis added)...”. Since Rodgers qualifies under §102(a), Rodgers cannot be excluded under §103(c), which requires that Rodgers qualify only under §102(e), (f), or (g).

Conclusion

20. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

21. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Art Unit: 2183

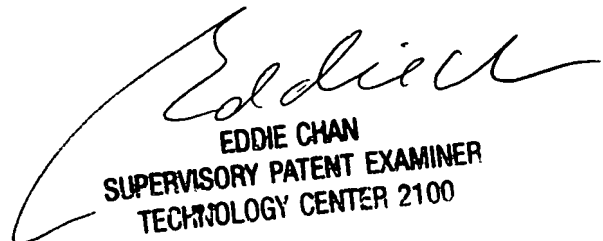
22. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aimee J. Li whose telephone number is (571) 272-4169. The examiner can normally be reached on M-T 7:00am-4:30pm.

23. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

24. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Aimee J Li
Examiner
Art Unit 2183

30 June 2007


EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100